Matlab优化

主讲人：饶志欢

现代优化算法

什么是优化？就是从各种方案中选取一个最好的。从数学角度看，优化理论就是研究如何在状态空间中寻找到全局最优点。

比如水泥混凝土的性能，涉及到水、沙、石子、水泥和其他掺杂物比例。学校课程表排课问题、售票员上岗问题、公司内部人员安排出效益等。降低成本、提高效益是问题的关键。

1.1 MATLAB解优化问题的主要函数

类　　型	模　　　型	基本函数名
一元函数极小	Min F (x) s.t.x1<x<x2	x=fminbnd('F', x_1, x_2)
无约束极小	Min F(X)	X=fminunc('F', X_0) X=fminsearch('F', X_0)
线性规划	Min $c^T X$ s.t.AX<=b	X=linprog(c, A, b)
二次规划	Min $\frac{1}{2}x^T Hx+c^T x$ s.t. Ax<=b	X=quadprog(H, c, A, b)
约束极小（非线性规划）	Min F(X) s.t. G(X) <=0	X=fmincon('FG', X_0)
达到目标问题	Min r s.t. F(x)-wr <=goal	X=fgoalattain('F', x, goal, w)
极小极大问题	Min max $\{F_i(x)\}$ x $\{F_i(x)\}$ s.t. G(x) <=0	X=fminimax('FG', x_0)

1.2 优化函数的输入变量

变量	描　　述	调用函数
f	线性规划的目标函数f*X 或二次规划的目标函数X'*H*X+f*X 中线性项的系数向量	linprog, quadprog
fun	非线性优化的目标函数.fun必须为行命令对象或M文件、嵌入函数、或MEX文件的名称	fminbnd, fminsearch, fminunc, fmincon, lsqcurvefit, lsqnonlin, fgoalattain, fminimax
H	二次规划的目标函数X'*H*X+f*X 中二次项的系数矩阵	quadprog
A, b	A矩阵和b向量分别为线性不等式约束：$AX \le b$中的系数矩阵和右端向量	linprog, quadprog, fgoalattain, fmincon, fminimax
Aeq, beq	Aeq矩阵和beq向量分别为线性等式约束：$Aeq \cdot X = beq$中的系数矩阵和右端向量	linprog, quadprog, fgoalattain, fmincon, fminimax
vlb, vub	X的下限和上限向量：vlb≤X≤vub	linprog, quadprog, fgoalattain, fmincon, fminimax, lsqcurvefit, lsqnonlin
X_0	迭代初始点坐标	除fminbnd外所有优化函数
x_1, x_2	函数最小化的区间	fminbnd
options	优化选项参数结构，定义用于优化函数的参数	所有优化函数

1.3 优化函数的输出变量下表

变量	描　　述	调用函数
x	由优化函数求得的值.若exitflag>0,则x为解;否则,x不是最终解,它只是迭代制止时优化过程的值	所有优化函数
fval	解x处的目标函数值	linprog, quadprog, fgoalattain, fmincon, fminimax, lsqcurvefit, lsqnonlin, fminbnd
exitflag	描述退出条件: ● exitflag>0,表目标函数收敛于解x处 ● exitflag=0,表已达到函数评价或迭代的最大次数 ● exitflag<0,表目标函数不收敛	
output	包含优化结果信息的输出结构. ● Iterations:迭代次数 ● Algorithm:所采用的算法 ● FuncCount:函数评价次数	所有优化函数

1.4 控制参数options的设置

(1)**Display**: 显示水平.取值为’off’时,不显示输出; 取值为’iter’时,显示每次迭代的信息;取值为’final’时,显示最终结果.默认值为’final’.

(2)**MaxFunEvals**: 允许进行函数评价的最大次数,取值为正整数.

(3) **MaxIter**: 允许进行迭代的最大次数,取值为正整数

控制参数options可以通过函数optimset创建或修改。命令的格式如下：

(1) **options=optimset(‘optimfun’)**

创建一个含有所有参数名,并与优化函数optimfun相关的默认值的选项结构options.

1.4 控制参数options的设置

(2)**options=optimset('param1',value1,'param2',value2,...)**

创建一个名称为options的优化选项参数,其中指定的参数具有指定值,所有未指定的参数取默认值.

(3)**options=optimset(oldops,'param1',value1,'param2',value2,...)**

创建名称为oldops的参数的拷贝,用指定的参数值修改oldops中相应的参数.

例：opts=optimset('Display','iter','TolFun',1e-8)

该语句创建一个称为opts的优化选项结构,其中显示参数设为'iter', TolFun参数设为1e-8.

2.1 一元函数无约束优化问题

一元函数无约束优化问题 $\min f(x), x_1 \le x \le x_2$

常用格式如下：

（1）**x= fminbnd (*fun,x1,x2*)**

（2）**x= fminbnd (*fun,x1,x2*， options)**

（3）**[x， fval]= fminbnd（...）**

（4）**[x， fval， exitflag]= fminbnd（...）**

（5）**[x， fval， exitflag， output]= fminbnd（...）**

函数fminbnd的算法基于黄金分割法和二次插值法，它要求目标函数必须是连续函数，并可能只给出局部最优解。

2.1 一元函数无约束优化问题

例 $f=2e^{-x}\sin x$ 在0<x<8中的最小值与最大值

程序如下：

```
f='2*exp(-x).*sin(x)';
fplot(f,[0,8]); %作图语句
[xmin,ymin]=fminbnd (f, 0,8)
f1='-2*exp(-x).*sin(x)';
[xmax,ymax]=fminbnd (f1, 0,8)
```

运行结果：

xmin = 3.9270 ymin = -0.0279

xmax = 0.7854 ymax = 0.6448

2.2 多元函数无约束优化问题

标准型为： *min F(X)*

命令格式为:

（1） x= fminunc（*fun,X0*）

或x=fminsearch（*fun,X0*）

（2） x= fminunc（*fun,X0*， options）；

或x=fminsearch（*fun,X0*， options）

（3） [x， fval]= fminunc（...）；

或[x， fval]= fminsearch（...）

（4） [x， fval， exitflag]= fminunc（...）；

或[x， fval， exitflag]= fminsearch

（5） [x， fval， exitflag， output]= fminunc（...）；

或[x， fval， exitflag， output]= fminsearch（.

3 多元函数无约束优化问题

说明:

• **fminsearch是用单纯形法寻优. fminunc的算法见以下几点说明:**

[1] fminunc为无约束优化提供了大型优化和中型优化算法。由options中的参数LargeScale控制：

LargeScale='on'(默认值),使用大型算法

LargeScale='off'(默认值),使用中型算法

[2] fminunc为中型优化算法的搜索方向提供了4种算法，由options中的参数HessUpdate控制：

HessUpdate='bfgs'（默认值），拟牛顿法的BFGS公式；

HessUpdate='dfp'，拟牛顿法的DFP公式；

HessUpdate='steepdesc'，最速下降法

[3] fminunc为中型优化算法的步长一维搜索提供了两种算法， 由options中参数LineSearchType控制：

LineSearchType='quadcubic'(缺省值)，混合的二次和三次多项式插值；

LineSearchType='cubicpoly'，三次多项式插

使用fminunc和 fminsearch可能会得到局部最优解

2.2 多元函数无约束优化问题

例 min f(x)=(4*x1^2+2*x2^2+4*x1*x2+2*x2+1)*exp(x1)

1、编写M-文件 fun1.m:

```
function f = fun1 (x)
f = exp(x(1))*(4*x(1)^2+2*x(2)^2+4*x(1)*x(2)+2*x(2)+1);
```

2、输入M文件wliti3.m如下:

```
x0 = [-1, 1];
x=fminunc('fun1',x0);
y=fun1(x)
```

3、运行结果:

x= 0.5000 -1.0000

y = 1.3029e-10

2.3 二次规划

标准型为:

$$\text{Min } Z = \frac{1}{2} X^T H X + c^T X$$

$$\text{s.t. } AX \le b \quad Aeq \cdot X = beq$$

$$VLB \le X \le VUB$$

用MATLAB软件求解,其输入格式如下:

1. x=quadprog(H,C,A,b);
2. x=quadprog(H,C,A,b,Aeq,beq);
3. x=quadprog(H,C,A,b,Aeq,beq,VLB,VUB);
4. x=quadprog(H,C,A,b, Aeq,beq ,VLB,VUB,X0);
5. x=quadprog(H,C,A,b, Aeq,beq ,VLB,VUB,X0,options);
6. [x,fval]=quaprog(...);
7. [x,fval,exitflag]=quaprog(...);
8. [x,fval,exitflag,output]=quaprog(...);

2.3 二次规划

例 **min f(x1,x2)=-2*x1-6*x2+x1^2-2*x1*x2+2*x2^2**

s.t. x1+x2≤2

-x1+2x2≤2

x1≥0, x2≥0

1、写成标准形式:

$$\min z = (x_1, x_2)\begin{pmatrix} 1 & -1 \\ -1 & 2 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} -2 \\ -6 \end{pmatrix}^T \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 \\ -1 & 2 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \le \begin{pmatrix} 2 \\ 2 \end{pmatrix}$$

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \le \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

2、输入命令:

H=[1 -1; -1 2];
c=[-2 ;-6];A=[1 1; -1 2];b=[2;2];
Aeq=[];beq=[]; VLB=[0;0];VUB=[];
[x,z]=quadprog(H,c,A,b,Aeq,beq,VLB,VUB)

3、运算结果为:

x =0.6667 1.3333 z = -8.2222

2.4 一般非线性规划

标准型为 *min F(X)*

s.t AX<=b *G(X)*

Ceq(X)=0 VLB X VUB

其中*X*为*n*维变元向量，*G(X)*与*Ceq(X)*均为非线性函数组成的向量，其它变量的含义与线性规划、二次规划中相同.用Matlab求解上述问题，基本步骤分三步：

1. 首先建立M文件*fun.m,*定义目标函数F（X）：

function f=fun(X);

f=F(X);

2. 若约束条件中有非线性约束:*G(X)* 或*Ceq(X)=0,*则建立M文件nonlcon.m定义函数G(X)与Ceq(X):

function [G,Ceq]=nonlcon(X)

G=...

Ceq=...

2.4 一般非线性规划

3. 建立主程序.非线性规划求解的函数是fmincon,命令的基本格式如下：

(1) *x=*fmincon*('fun',X0,A,b)*

(2) *x=*fmincon*('fun',X0,A,b,Aeq,beq)*

(3) *x=*fmincon*('fun',X0,A,b, Aeq,beq,VLB,VUB)*

(4) *x=*fmincon*('fun',X0,A,b,Aeq,beq,VLB,VUB,'nonlcon')*

(5)*x=*fmincon*('fun',X0,A,b,Aeq,beq,VLB,VUB,'nonlcon',options)*

(6) *[x,fval]=* fmincon(...)

(7) *[x,fval,exitflag]=* fmincon(...)

(8)[x,fval,exitflag,output]= fmincon(...)

2.4 一般非线性规划

注意：

[1] fmincon函数提供了大型优化算法和中型优化算法。默认时，若在fun函数中提供了梯度（options参数的GradObj设置为'on'），并且只有上下界存在或只有等式约束，fmincon函数将选择大型算法。当既有等式约束又有梯度约束时，使用中型算法。

[2] fmincon函数的中型算法使用的是序列二次规划法。在每一步迭代中求解二次规划子问题，并用BFGS法更新拉格朗日Hessian矩阵。

[3] fmincon函数可能会给出局部最优解，这与初值*X0*的选取有关。

2.4 一般非线性规划

例

$$\min f = -x_1 - 2x_2 + \frac{1}{2}x_1^2 + \frac{1}{2}x_2^2$$

$$2x_1 + 3x_2 \le 6$$

$$x_1 + 4x_2 \le 5$$

$$x_1, x_2 \ge 0$$

1、**写成标准形式：**

$$\min f = -x_1 - 2x_2 + \frac{1}{2}x_1^2 + \frac{1}{2}x_2^2$$

$$\begin{pmatrix} 2x_1 + 3x_2 - 6 \\ x_1 + 4x_2 - 5 \end{pmatrix} \le \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} < \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

2.4 一般非线性规划

2、先建立**M-**文件 **fun3.m:**

```
function f=fun3(x);
f=-x(1)-2*x(2)+(1/2)*x(1)^2+(1/2)*x(2)^2
```

3、再建立主程序youh2.m：

```
x0=[1;1];
A=[2 3 ;1 4]; b=[6;5];
Aeq=[];beq=[];
VLB=[0;0]; VUB=[];
[x,fval]=fmincon('fun3',x0,A,b,Aeq,beq,VLB,VUB)
```

4、**运算结果为：**

x = 0.7647 1.0588

fval = -2.0294

2.5 线性规划问题

线性规划问题

线性规划问题是目标函数和约束条件均为线性函数的问题，MATLAB6.0 解决的线性规划问题的标准形式为：

min f(x)

sub.to：

x A ≤b · x Aeq = beq· ub≤ x≤ lb

其中 f、x、b、beq、lb、ub 为向量，A、Aeq 为矩阵。其它形式的线性规划问题都可经过适当变换化为此标准形式。

x = linprog(f,A,b,Aeq,beq,lb,ub,x0) %设置初值 x0

3 MATLAB优化工具箱

1 工具箱

（1）求解无约束条件非线性极小值；

（2）求解约束条件下非线性极小值，包括目标逼近　　问题、极大-极小值问题和半无限极小值问题；

（3）求解二次规划和线性规划问题；

（4）非线性最小二乘逼近和曲线拟合；

（5）非线性系统的方程求解；

（6）约束条件下的线性最小二乘优化；

（7）求解复杂结构的大规模优化问题。

3 MATLAB优化工具箱

2 工具箱函数

一元函数极小值　　X=fminbnd(‘F’,x1,x2)

无约束极小　　X=fminunc(‘F’,X0)
X=fminsearch(‘F’,X0)

线性规划　　X=linprog(c,A,b)

0-1整数规划　　X=bintprog(F)

二次规划　　X=quadprog(H,c,A,b)

约束极小值（非线性规划）
X=fmincon(‘FG’,X0)

非线性最小二乘　　X=lsqnonlin(F,X0)

目标达到问题　　X=fgoalattain(‘F’,x,goal,w)

极小极大问题　　X=fminimax(‘FG’,x0)

3.1 GUI优化工具

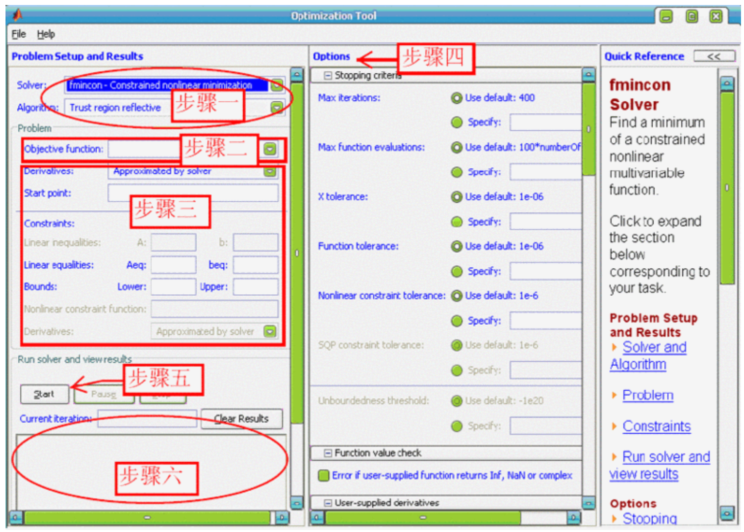
命令行输入optimtool；

Start->Toolboxes->Optimization->Optimization tool(optimtool)

3.2 使用步骤

1. 选择求解器solver和优化算法algorithm；
2. 选定目标函数（objective function）；
3. 设定目标函数的相关参数；
4. 设置优化选项；
5. 单击“start”按钮，运行求解；
6. 查看求解器的状态和求解结果；
7. 将目标函数、选项和结果导入\导出。

3.2 使用步骤



3.3 应用实例一

无约束优化（fminunc求解器）

求f(x)=x^2+4*x-6极小值，初始点取x=0。

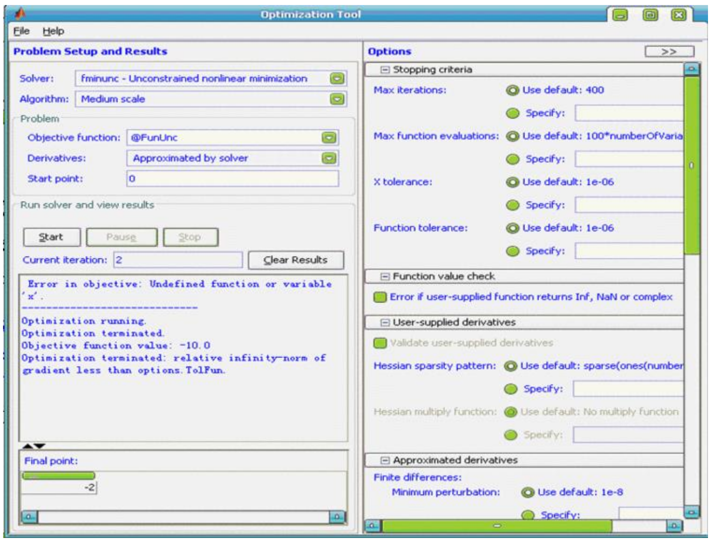
解：

1. 首先建立目标函数文件FunUnc.m文件：

```
function y=FunUnc(x)
y=x^2+4*x-6;
```

2. 然后启动优化工具

3.3 应用实例一



3.3 应用实例一

Algorithm有两个选择：Large scale和Medium scale， 设置完参数点击start即可得到上图中的结果。

3.4 应用实例二

无约束优化（fminsearch求解器）

求f(x)=|x^2-3*x+2|的极小值，初始点取x=-7

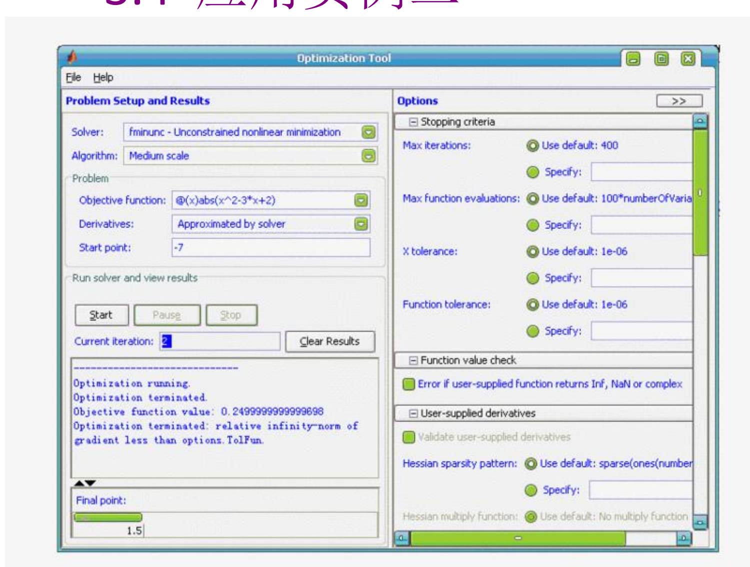
解：

1.解：启动优化工具；

用fminunc时设置参数如图

2. 运行得到结果

3.4 应用实例二

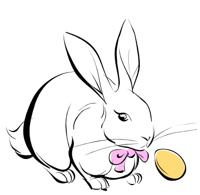


4 高级优化算法

习惯上，将优化算法分为两类：**局部优化算法**和**全局性优化算法**。前者可以称为**经典优化算法**，已经得到了人们广泛深入的研究。线性规划、整数规划、0–1规划、非线性规划、排队论、决策论。后者习惯上称为**现代优化算法**，是20世纪80年代兴起的新型全局性优化算法，主要包括**禁忌搜索**、**模拟退火**、遗传算法、神经网络等，其主要应用对象是优化问题中的**难解问题**，即**NP–hard**问题

4.1 算法简介

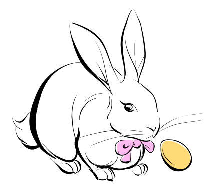
为了找出地球上最高的山，一群有志气的兔子们开始想办法。



4.1 算法简介

方案一：兔子们吃了失忆药片，并被发射到太空，然后随机落到了地球上的某些地方。他们不知道自己的使命是什么。但是，如果你过几年就杀死一部分海拔低的兔子，多产的兔子们自己就会找到珠穆朗玛峰。

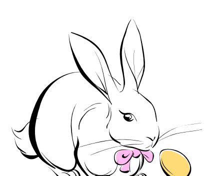
遗传算法



4.1 算法简介

方案二：兔子们朝着比现在高的地方跳去，它们找到了不远处的最高山峰。但是这座山不一定是珠穆朗玛峰。其实，它们这种做法只是自己心理上认为找到了最高的山，并不能保证局部最优值就是全局最优值。

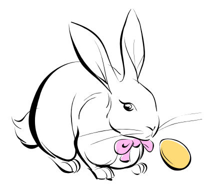
局部搜索法



4.1 算法简介

方案三：兔子们知道一个兔子的力量是渺小的。于是，它们互相转告着，哪里的山已经找过，并且找过的每一座山他们都留下一只兔子做记号。这样，它们制定了下一步去哪里寻找的策略。

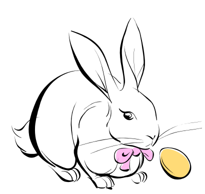
禁忌搜索法



4.1 算法简介

方案四：兔子们用酒将自己灌醉了。它们随机地跳了很长时间。在这期间，它们可能走向高处，也可能踏入平地。但是，随着时间的流逝，它们渐渐清醒了并朝最高方向跳去。

模拟退火法



4.2 遗传算法基本思想

遗传算法流程图如下

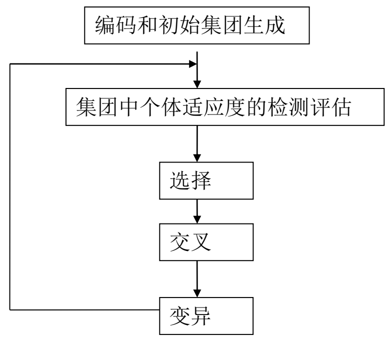


图1 遗传算法的基本流程

4.3 模拟退火算法简介

1) 任选一初始状态　作为初始解，并设初始温度；
2) 调用采样算法，然后返回到当前解；
3) 按一定的方式将降温
4) 检查退火过程是否结束，否则转到 2);
5) 以当前解　作为最优解输出。